Iterative Algorithms for Polynomial Eigenvalue Decomposition and Applications

Stephan Weiss

UDRC — Loughborough, Surrey, Strathclyde & Cardiff Consortium
Department of Electonic & Electrical Engineering
University of Strathclyde, Glasgow, Scotland, UK

UDRC Summer School, Heriot-Watt University, 23-27 June 2014

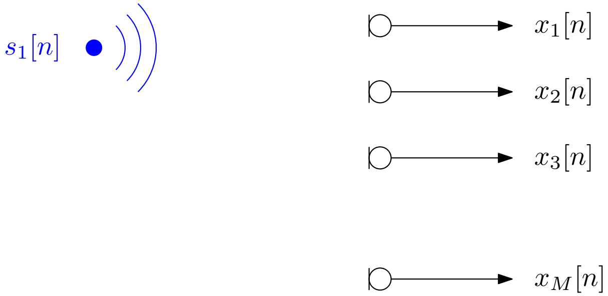
With many thanks to:
J.G. McWhirter, S. Redif, C.H. Ta, W. Al-Hanafy, S. Lambotharan, M. Alrmah, J. Corr, K. Thompson and I.K. Proulder

Presentation Overview

1. Overview
2. Eigenvalue (EVD) and singular value decomposition (SVD)
3. Narrowband source separation
4. Broadband problem
5. Polynomial matrices and basic operations
6. Polynomial eigenvalue decomposition algorithms
   - 6.1 sequential best rotation (SBR2)
   - 6.2 sequential matrix diagonalisation (SMD)
7. Applications
   - 7.1 broadband / polynomial subspace decomposition
   - 7.2 polynomial MUSIC
8. Summary

Narrowband Source Model

- Scenario with sensor array and far-field sources:

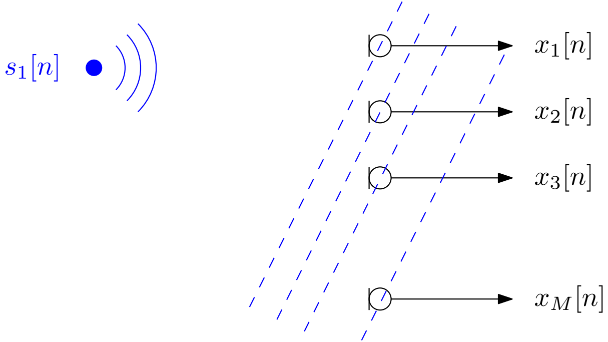


- for the narrowband case, the source signals arrive with delays, expressed by phase shifts in a steering vector
- data model:

$$\mathbf{x}[n] =$$

Narrowband Source Model

- Scenario with sensor array and far-field sources:

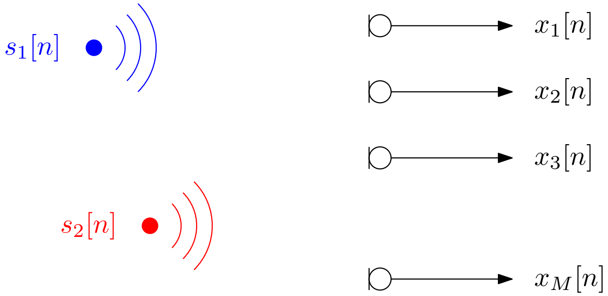


- for the narrowband case, the source signals arrive with delays, expressed by phase shifts in a steering vector $\mathbf{s}_1$
- data model:

$$\mathbf{x}[n] = s_1[n] \cdot \mathbf{s}_1$$

Narrowband Source Model

- Scenario with sensor array and far-field sources:

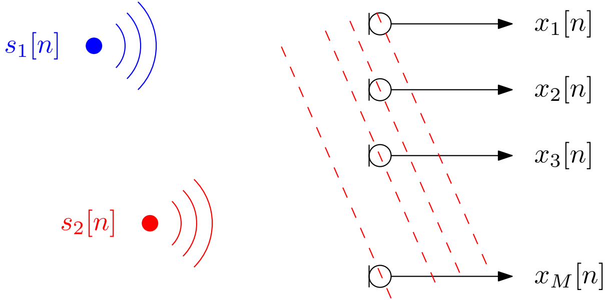


- for the narrowband case, the source signals arrive with delays, expressed by phase shifts in a steering vector $\mathbf{s}_1$
- data model:

$$\mathbf{x}[n] = s_1[n] \cdot \mathbf{s}_1$$

Narrowband Source Model

- Scenario with sensor array and far-field sources:

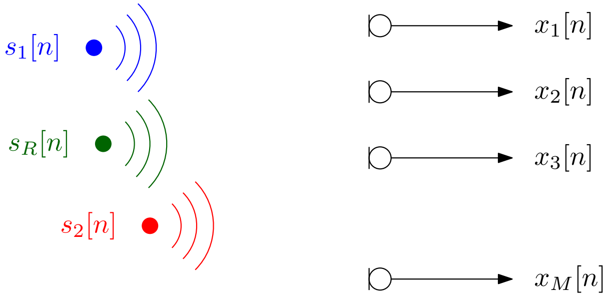


- for the narrowband case, the source signals arrive with delays, expressed by phase shifts in a steering vector $\mathbf{s}_1$, $\mathbf{s}_2$
- data model:

$$\mathbf{x}[n] = s_1[n] \cdot \mathbf{s}_1 + s_1[n] \cdot \mathbf{s_2}$$

Narrowband Source Model

- Scenario with sensor array and far-field sources:

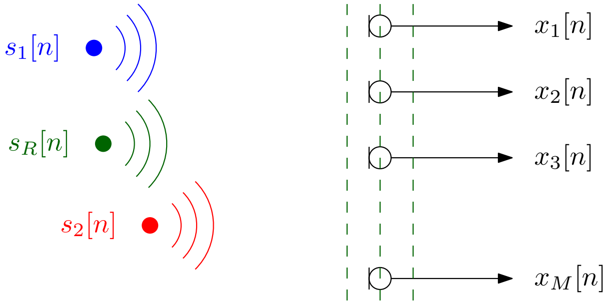


- for the narrowband case, the source signals arrive with delays, expressed by phase shifts in a steering vector $\mathbf{s}_1$, $\mathbf{s}_2$
- data model:

$$\mathbf{x}[n] = s_1[n] \cdot \mathbf{s}_1 + s_1[n] \cdot \mathbf{s_2}$$

Narrowband Source Model

- Scenario with sensor array and far-field sources:



- for the narrowband case, the source signals arrive with delays, expressed by phase shifts in a steering vector $\mathbf{s}_1$, $\mathbf{s}_2$, $\ldots \mathbf{s}_R$;
- data model:

$$\mathbf{x}[n] = s_1[n] \cdot \mathbf{s_1} + s_1[n] \cdot \mathbf{s_2} + \cdots + s_R[n] \cdot \mathbf{s}_R = \sum_{r=1}^{R} s_r[n] \cdot \mathbf{s}_r$$

University of Strathclyde Engineering

Steering Vector

- A signal $s[n]$ arriving at the array can be characterised by the delays of its wavefront (neglecting attenuation):

$$\begin{bmatrix} x_0[n] \\ x_1[n] \\ \vdots \\ x_{M-1}[n] \end{bmatrix} = \begin{bmatrix} s[n-\tau_0] \\ s[n-\tau_1] \\ \vdots \\ s[n-\tau_{M-1}] \end{bmatrix} = \begin{bmatrix} \delta[n-\tau_0] \\ \delta[n-\tau_1] \\ \vdots \\ \delta[n-\tau_{M-1}] \end{bmatrix} * s[n] \circ\!\!-\!\!\bullet\ \mathbf{a}_\vartheta(z) S(z)$$

- if evaluated at a narrowband normalised angular frequency Ω_i, the time delays τ_m in the broadband steering vector $\mathbf{a}_\vartheta(z)$ collapse to phase shifts in the narrowband steering vector $\mathbf{a}_{\vartheta,\Omega_i}$,

$$\mathbf{a}_{\vartheta,\Omega_i} = \mathbf{a}_\vartheta(z)|_{z=e^{j\Omega_i}} = \begin{bmatrix} e^{-j\tau_0\Omega_i} \\ e^{-j\tau_1\Omega_i} \\ \vdots \\ e^{-j\tau_{M-1}\Omega_i} \end{bmatrix} .$$

Data and Covariance Matrices

- A data matrix $\mathbf{X} \in \mathbb{C}^{M\times L}$ can be formed from L measurements:

$$\mathbf{X} = \left[\begin{array}{cccc} \mathbf{x}[n] & \mathbf{x}[n+1] & \ldots & \mathbf{x}[n+L-1] \end{array} \right]$$

- assuming that all $x_m[n]$, $m = 1, 2, \ldots M$ are zero mean, the (instantaneous) data covariance matrix is

$$\mathbf{R} = \mathcal{E}\{\mathbf{x}[n]\mathbf{x}^{\mathrm{H}}[n]\} \approx \frac{1}{L}\mathbf{X}\mathbf{X}^{\mathrm{H}}$$

  where the approximation assumes ergodicity and a sufficiently large L;
- Problem: can we tell from $\mathbf{X}$ or $\mathbf{R}$ (i) the number of sources and (ii) their orgin / time series?
- w.r.t. Jonathon Chamber's introduction, we here only consider the underdetermined case of more sensors than sources, $M \geq K$, and generally $L \gg M$.

SVD of Data Matrix

- Singular value decomposition of $\mathbf{X}$:

$$\mathbf{X} = \mathbf{U} \; \boldsymbol{\Sigma} \; \mathbf{V}^{\mathrm{H}}$$

- unitary matrices $\mathbf{U} = [\mathbf{u}_1 \ldots \mathbf{u}_M]$ and $\mathbf{V} = [\mathbf{v}_1 \ldots \mathbf{v}_L]$;
- diagonal $\boldsymbol{\Sigma}$ contains the real, positive semidefinite singular values of $\mathbf{X}$ in descending order:

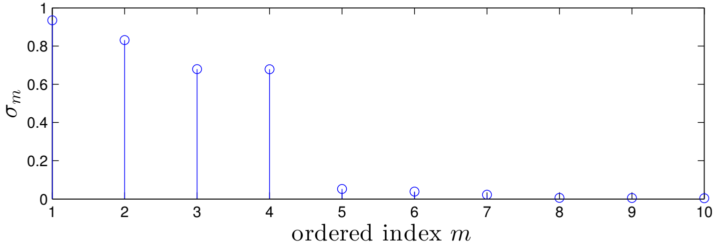
$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1 & 0 & \ldots & 0 & 0 & \ldots & 0 \\ 0 & \sigma_2 & \ddots & \vdots & \vdots & & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots & & \vdots \\ 0 & & 0 & \sigma_M & 0 & \ldots & 0 \end{bmatrix}$$

with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_M \geq 0$.

University of Strathclyde Engineering

Singular Values

- If the array is illuminated by $R \leq M$ linearly independent sources, the rank of
  the data matrix is
  $$\text{rank}\{\mathbf{X}\} = R$$
- only the first R singular values of $\mathbf{X}$ will be non-zero;
- in practice, noise often will ensure that $\text{rank}\{\mathbf{X}\} = M$, with $M - R$ trailing singular values that define the noise floor:



- therefore, by thresholding singular values, it is possible to estimate the number of linearly independent sources R.

Subspace Decomposition

- If $\text{rank}\{\mathbf{X}\} = R$, the SVD can be split:

$$\mathbf{X} = [\mathbf{U}_s \;\; \mathbf{U}_n] \begin{bmatrix} \boldsymbol{\Sigma}_s & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_n \end{bmatrix} \begin{bmatrix} \mathbf{V}_s^{\mathrm{H}} \\ \mathbf{V}_n^{\mathrm{H}} \end{bmatrix}$$

- with $\mathbf{U}_s \in \mathbb{C}^{M\times R}$ and $\mathbf{V}_s^{\mathrm{H}} \in \mathbb{C}^{R\times L}$ corresponding to the R largest singular values;
- $\mathbf{U}_s$ and $\mathbf{V}_s^{\mathrm{H}}$ define the signal-plus-noise subspace of $\mathbf{X}$:

$$\mathbf{X} = \sum_{m=1}^{M} \sigma_m \mathbf{u}_m \mathbf{v}_m^{\mathrm{H}} \approx \sum_{m=1}^{R} \sigma_m \mathbf{u}_m \mathbf{v}_m^{\mathrm{H}}$$

- the complements $\mathbf{U}_n$ and $\mathbf{V}_n^{\mathrm{H}}$,

$$\mathbf{U}_s^{\mathrm{H}}\mathbf{U}_n = \mathbf{0} \qquad , \qquad \mathbf{V}_s\mathbf{V}_n^{\mathrm{H}} = \mathbf{0}$$

define the noise-only subspace of $\mathbf{X}$.

SVD via Two EVDs

- Any Hermitian matrix $\mathbf{A} = \mathbf{A}^{\mathrm{H}}$ allows an eigenvalue decomposition

$$\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{\mathrm{H}}$$

  with $\mathbf{Q}$ unitary and the eigenvalues in $\mathbf{\Lambda}$ real valued and positive semi-definite;
- postulating $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}}$, therefore:

$$\mathbf{X}\mathbf{X}^{\mathrm{H}} = (\mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}})(\mathbf{V}\mathbf{\Sigma}^{\mathrm{H}}\mathbf{U}^{\mathrm{H}}) = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\mathrm{H}} \quad (1)$$

$$\mathbf{X}^{\mathrm{H}}\mathbf{X} = (\mathbf{V}\mathbf{\Sigma}^{\mathrm{H}}\mathbf{U}^{\mathrm{H}})(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}}) = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{\mathrm{H}} \quad (2)$$

- (ordered) eigenvalues relate to the singular values: $\lambda_m = \sigma_m^2$;
- the covariance matrix $\mathbf{R} = \frac{1}{L}\mathbf{X}\mathbf{X}$ has the same rank as the data matrix $\mathbf{X}$, and with $\mathbf{U}$ provides access to the same spatial subspace decomposition.

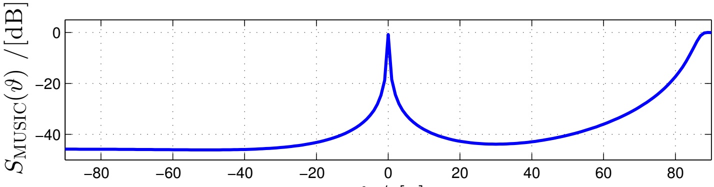
Narrowband MUSIC Algorithm

- EVD of the narrowband covariance matrix identifies signal-plus-noise and noise-only subspaces

$$\mathbf{R} = [\mathbf{U}_s \;\; \mathbf{U}_n] \begin{bmatrix} \mathbf{\Lambda}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda}_n \end{bmatrix} \begin{bmatrix} \mathbf{U}_s^{\mathrm{H}} \\ \mathbf{U}_n^{\mathrm{H}} \end{bmatrix}$$

- scanning the signal-plus-noise subspace could only help to retrieve sources with orthogonal steering vectors;
- therefore, the multiple signal classification (MUSIC) algorithm scans the noise-only subspace for minima, or maxima of its reciprocal

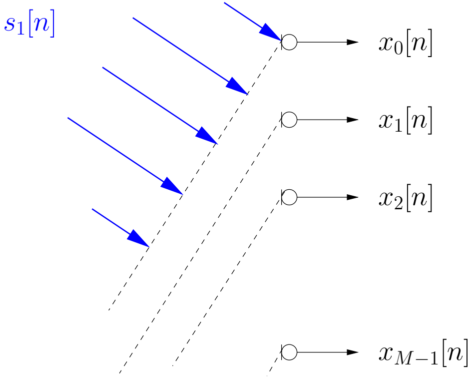
$$S_{\mathrm{MUSIC}}(\vartheta) = \frac{1}{\|\mathbf{U}_n \mathbf{a}_{\vartheta,\Omega_i}\|_2^2}$$



Narrowband Source Separation

- Via SVD of the data matrix $\mathbf{X}$ or EVD of the covariance matrix $\mathbf{R}$, we can determine the number of linearly independent sources R;
- using the subspace decompositions offered by EVD/SVD, the directions of arrival can be estimated using e.g. MUSIC;
- based on knowledge of the angle of arrival, beamforming could be applied to $\mathbf{X}$ to extract specific sources;
- overall: EVD (and SVD) can play a vital part in narrowband source separation;
- what about broadband source separation?

Broadband Array Scenario



- Compared to the narrowband case, time delays rather than phase shifts bear information on the direction of a source.

Broadband Steering Vector

- A signal $s[n]$ arriving at the array can be characterised by the delays of its wavefront (neglecting attenuation):

$$\begin{bmatrix} x_0[n] \\ x_1[n] \\ \vdots \\ x_{M-1}[n] \end{bmatrix} = \begin{bmatrix} s[n-\tau_0] \\ s[n-\tau_1] \\ \vdots \\ s[n-\tau_{M-1}] \end{bmatrix} = \begin{bmatrix} \delta[n-\tau_0] \\ \delta[n-\tau_1] \\ \vdots \\ \delta[n-\tau_{M-1}] \end{bmatrix} * s[n] \circ\!\!-\!\!\bullet\ \mathbf{a}_\vartheta(z) S(z)$$

- if evaluated at a narrowband normalised angular frequency Ω_i, the time delays τ_m in the broadband steering vector $\mathbf{a}_\vartheta(z)$ collapse to phase shifts in the narrowband steering vector $\mathbf{a}_{\vartheta,\Omega_i}$,

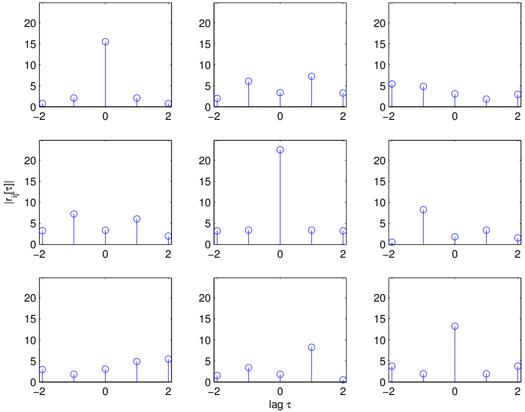
$$\mathbf{a}_{\vartheta,\Omega_i} = \mathbf{a}_\vartheta(z)|_{z=e^{j\Omega_i}} = \begin{bmatrix} e^{-j\tau_0\Omega_i} \\ e^{-j\tau_1\Omega_i} \\ \vdots \\ e^{-j\tau_{M-1}\Omega_i} \end{bmatrix} .$$

Space-Time Covariance Matrix

- If delays must be considered, the (space-time) covariance matrix must capture the lag τ:

$$\mathbf{R}[\tau] = \mathcal{E}\{\mathbf{x}[n] \cdot \mathbf{x}^{\mathrm{H}}[n-\tau]\}$$

- $\mathbf{R}[\tau]$ contains auto- and cross-correlation sequences:



Cross Spectral Density Matrix

- z-transform of the space-time covariance matrix is given by

$$\mathbf{R}[\tau] = \mathcal{E}\{\mathbf{x}_n \mathbf{x}_{n-\tau}^{\mathrm{H}}\} \quad \circ\!\!-\!\!\bullet \quad \boldsymbol{R}(z) = \sum_l S_l(z)\mathbf{a}_{\vartheta_l}(z)\tilde{\mathbf{a}}_{\vartheta_l}(z) + \sigma_N^2 \mathbf{I}$$

  with ϑ_l the direction of arrival and $S_l(z)$ the PSD of the lth source;
- $\boldsymbol{R}(z)$ is the cross spectral density (CSD) matrix;
- the instantaneous covariance matrix (no lag parameter τ)

$$\mathbf{R} = \mathcal{E}\{\mathbf{x}_n \mathbf{x}_n^{\mathrm{H}}\} = \mathbf{R}[0]$$

CSD Matrix Properties

- The CSD matrix $\boldsymbol{R}(z)$ is a matrix polynomial or polynomial with matrix-valued coefficients:

$$\boldsymbol{R}(z) = \cdots + \mathbf{R}_{-2}z^{2} + \mathbf{R}_{-1}z^{1} + \mathbf{R}_{0} + \mathbf{R}_{1}z^{-1} + \mathbf{R}_{2}z^{-2} + \ldots$$

- the symmetry of the cross-correlation sequences $r_{xy}[\tau] = r^{*}_{yx}[-\tau]$ is reflected in the CSD matrix $\boldsymbol{R}(z)$:

$$\mathbf{R}_{\tau} = \mathbf{R}^{\mathrm{H}}_{-\tau}$$

- therefore with the parahermitian operator $\tilde{\{\cdot\}}$

$$\boldsymbol{R}(z) = \boldsymbol{R}^{\mathrm{H}}(z^{-1}) = \tilde{\boldsymbol{R}}(z)$$

- a matrix fulfilling $\boldsymbol{R}(z) = \tilde{\boldsymbol{R}}(z)$ is called a parahermitian matrix.

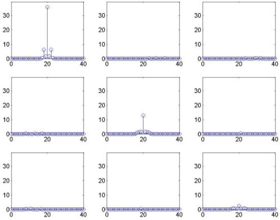
Polynomial Eigenvalue Decomposition

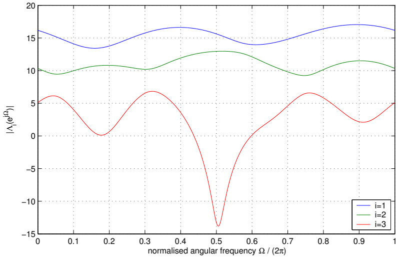
[McWhirter *et al.*, IEEE TSP 2007]

- Polynomial EVD of the CSD matrix

$$\boldsymbol{R}(z) = \boldsymbol{Q}(z)\ \boldsymbol{\Lambda}(z)\ \tilde{\boldsymbol{Q}}(z)$$

- with paraunitary $\mathbf{Q}(z)$, s.t. $\mathbf{Q}(z)\tilde{\mathbf{Q}}(z) = \mathbf{I}$;
- diagonalised and spectrally majorised $\boldsymbol{\Lambda}(z)$:





- $\boldsymbol{Q}(z)$ can be FIR of sufficiently high order [Icart & Comon 2012]

Iterative PEVD Algorithms

- Second order sequential best rotation (SBR2, McWhirter 2007);
- iterative approach based on an elementary paraunitary operation:

$$
\begin{aligned}
\boldsymbol{S}_0(z) &= \boldsymbol{R}(z) \\
&\vdots \\
\boldsymbol{S}_{i+1}(z) &= \tilde{\boldsymbol{H}}_{i+1}(z)\boldsymbol{S}_{i+1}(z)\boldsymbol{H}_{i+1}(z)
\end{aligned}
$$

- $\boldsymbol{H}_i(z)$ is an elementary paraunitary operation, which at the ith step eliminates the largest off-diagonal element in $\boldsymbol{s}_{i-1}(z)$;
- stop after L iterations:

$$
\hat{\boldsymbol{\Lambda}}(z) = \boldsymbol{S}_L(z) \qquad , \qquad \boldsymbol{Q}(z) = \prod_{i=1}^{L} \boldsymbol{H}_i(z)
$$

- sequential matrix diagonalisation (SMD) and
- multiple-shift SMD (MS-SMD) will follow the same scheme ...

Elementary Paraunitary Operation

- An elementary paraunitary matrix [Vaidyanathan] is defined as

$$\boldsymbol{H}_i(z) = \mathbf{I} - \mathbf{v}_i\mathbf{v}_i^{\mathrm{H}} + z^{-1}\mathbf{v}_i\mathbf{v}_i^{\mathrm{H}}$$

- we utilise a different definition:

$$\boldsymbol{H}_i(z) = \boldsymbol{D}_i(z)\mathbf{Q}_i$$

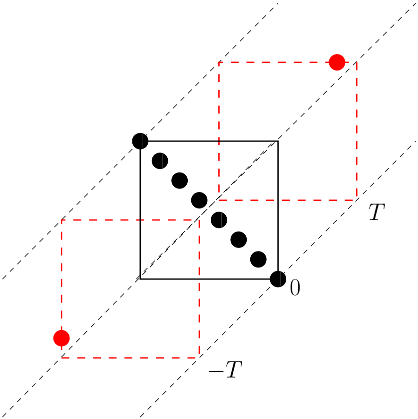
- $\boldsymbol{D}_i(z)$ is a delay matrix:

$$\boldsymbol{D}_i(z) = \mathrm{diag}\{1 \; \ldots \; 1 \; z^{-\tau} \; 1 \; \ldots \; 1\}$$

- $\mathbf{Q}_i(z)$ is a Givens rotation.

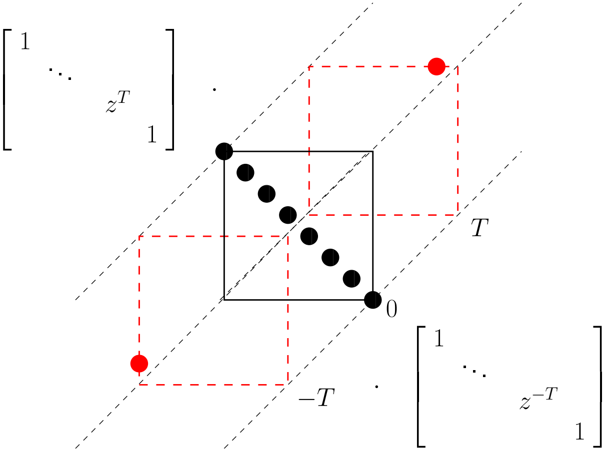
Sequential Best Rotation Algorithm (McWhirter)

- At iteration i, consider $\boldsymbol{S}_{i-1}(z) \circ\!\!-\!\!\bullet \mathbf{S}_{i-1}[\tau]$



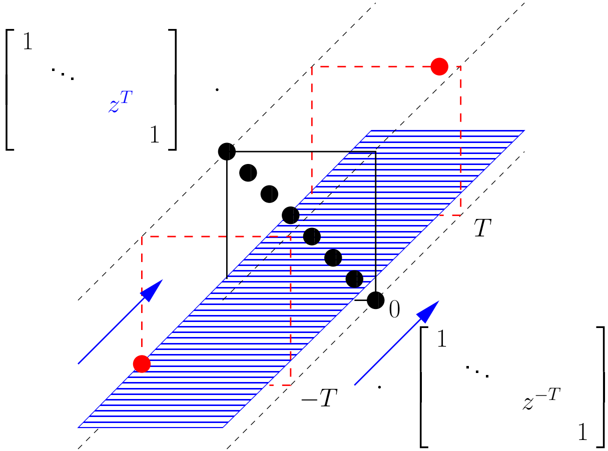
Sequential Best Rotation Algorithm (McWhirter)

- $\tilde{\boldsymbol{D}}_i(z)\boldsymbol{S}_{i-1}(z)\boldsymbol{D}_i(z)$



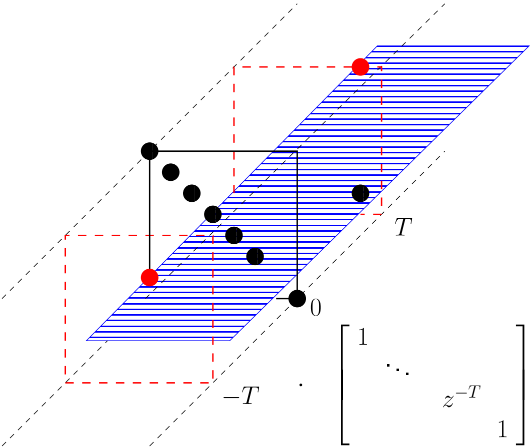
Sequential Best Rotation Algorithm (McWhirter)

- $\tilde{\boldsymbol{D}}_i(z)$ advances a row-slice of $\boldsymbol{S}_{i-1}(z)$ by T



Sequential Best Rotation Algorithm (McWhirter)

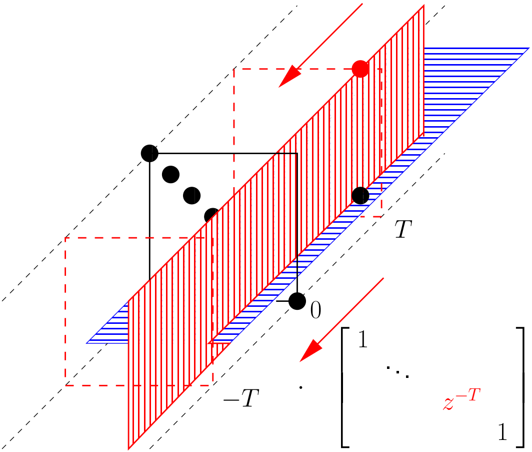
- the off-diagonal element at $-T$ has now been translated to lag zero



$$\cdot \begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & z^{-T} & \\ & & & 1 \end{bmatrix}$$

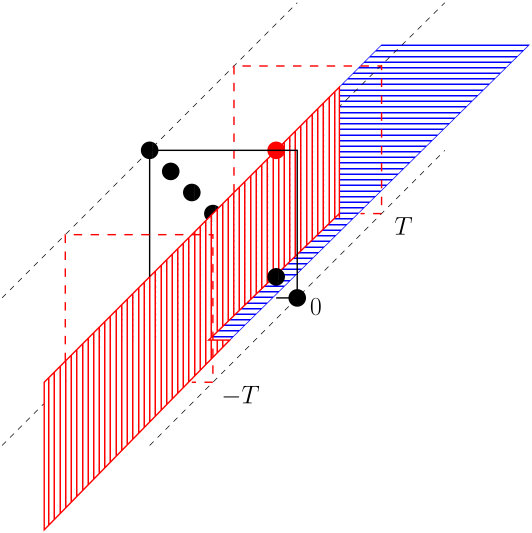
Sequential Best Rotation Algorithm (McWhirter)

- $\mathbf{D}_i(z)$ delays a column-slice of $\boldsymbol{S}_{i-1}(z)$ by T



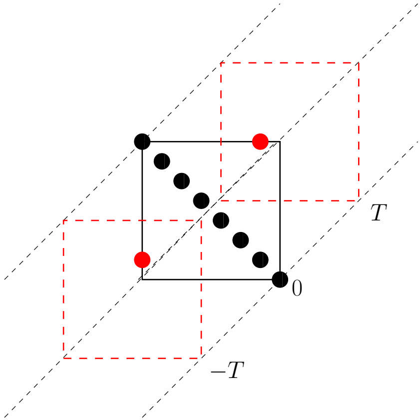
Sequential Best Rotation Algorithm (McWhirter)

- the off-diagonal element at $-T$ has now been translated to lag zero



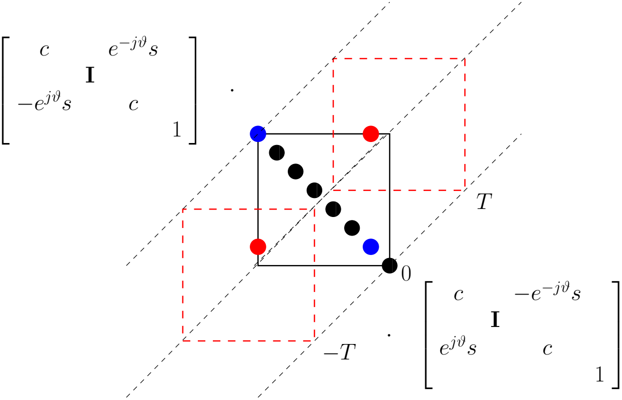
Sequential Best Rotation Algorithm (McWhirter)

- the step $\tilde{\boldsymbol{D}}_i(z)\boldsymbol{S}_{i-1}(z)\boldsymbol{D}_i(z)$ has brought the largest off-diagonal elements to lag 0.



Sequential Best Rotation Algorithm (McWhirter)

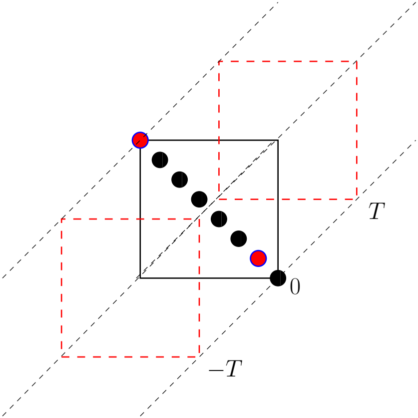
- Jacobi step to eliminate largest off-diagonal elements by $\mathbf{Q}_i$



Sequential Best Rotation Algorithm (McWhirter)

- iteration i is completed, having performed

$$\boldsymbol{S}_i(z) = \mathbf{Q}_i \boldsymbol{D}_i(z) \boldsymbol{S}_{i-1}(z) \tilde{\boldsymbol{D}}_i(z) \tilde{\mathbf{Q}}_i(z)$$



SBR2 Outcome

- At the ith iteration, the zeroing of off-diagonal elements achieved during previous steps may be partially undone;
- however, the algorithm has been shown to converge, transfering energy onto the main diagonal at every step (McWhirter 2007);
- after L iterations, we reach an approximate diagonalisation

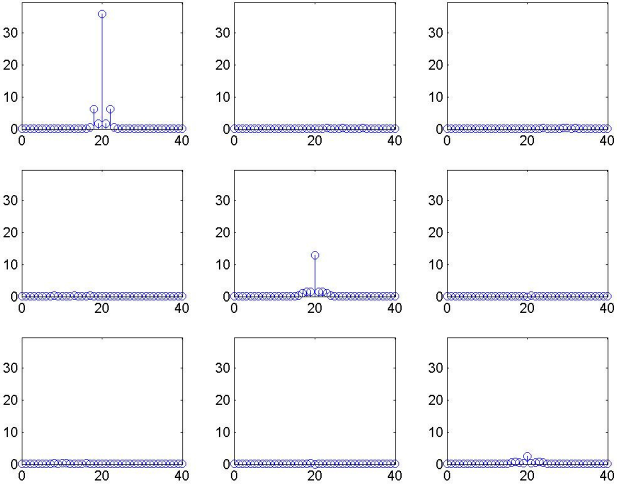
$$\hat{\mathbf{\Lambda}}(z) = \boldsymbol{S}^L(z) = \tilde{\boldsymbol{Q}}(z)\boldsymbol{R}(z)\boldsymbol{Q}(z)$$

with

$$\boldsymbol{Q}(z) = \prod_{i=1}^{L} \boldsymbol{D}_i(z)\mathbf{Q}_i$$

- diagonalisation of the previous 3×3 polynomial matrix . . .

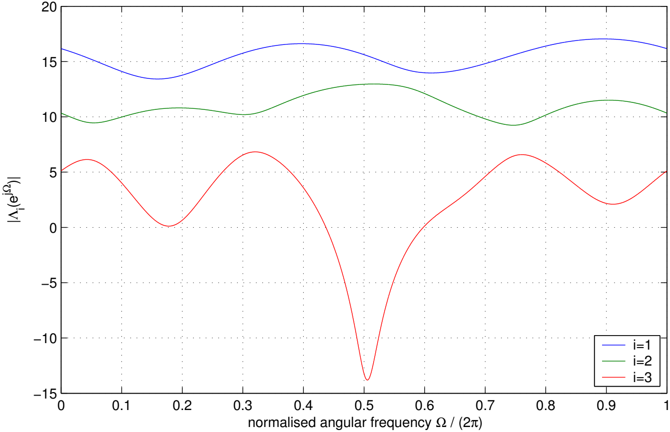
SBR2 Example — Diagonalisation



lag τ

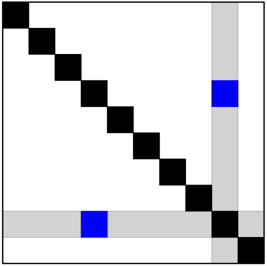
SBR2 Example — Spectral Majorisation

- The on-diagonal elements are spectrally majorised



SBR2 — Givens Rotation

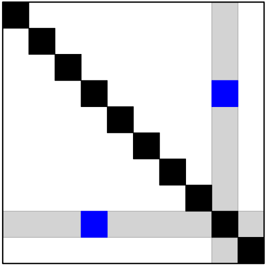
- A Givens rotation eliminates the maximum off-diagonal element once brought onto the lag-zero matrix;
- note I: in the lag-zero matrix, one column and one row are modified by the shift:

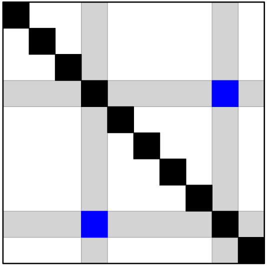


- note II: a Givens rotation only affects two columns and two rows in every matrix;
- Givens rotation is relatively low in computational cost!

SBR2 — Givens Rotation

- A Givens rotation eliminates the maximum off-diagonal element once brought onto the lag-zero matrix;
- note I: in the lag-zero matrix, one column and one row are modified by the shift:

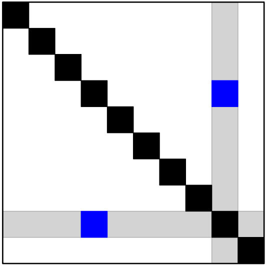




- note II: a Givens rotation only affects two columns and two rows in every matrix;
- Givens rotation is relatively low in computational cost!

Sequential Matrix Diagonalisation (SMD)

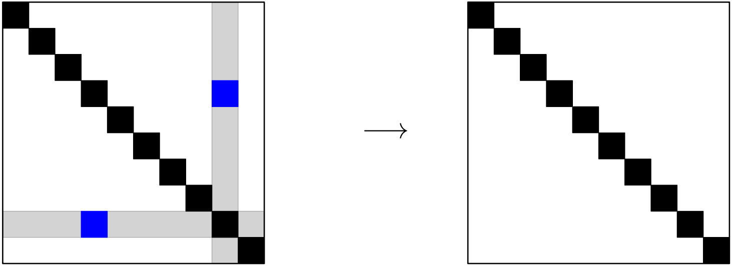
- Main idea — the zero-lag matrix is diagonalised in every step;
- initialisation: diagonalise $\mathbf{R}[0]$ by EVD and apply modal matrix to all matrix coefficients $\longrightarrow \boldsymbol{S}_0$;
- at the ith step as in SBR2, the maximum element (or column with max. norm) is shifted to the lag-zero matrix:



- an EVD is used to re-diagonalise the zero-lag matrix;
- a full modal matrix has to applied at all lags — more costly than SBR2.

Sequential Matrix Diagonalisation (SMD)

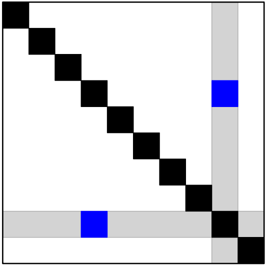
- Main idea — the zero-lag matrix is diagonalised in every step;
- initialisation: diagonalise $\mathbf{R}[0]$ by EVD and apply modal matrix to all matrix coefficients $\longrightarrow \boldsymbol{S}_0$;
- at the ith step as in SBR2, the maximum element (or column with max. norm) is shifted to the lag-zero matrix:



- an EVD is used to re-diagonalise the zero-lag matrix;
- a full modal matrix has to applied at all lags — more costly than SBR2.

Multiple Shift SMD (SMD)

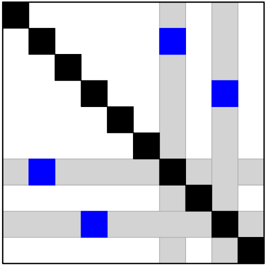
- SMD converges faster than SBR2 — more energy is transfered per iteration step;
- SMD is more expensive than SBR2 — full matrix multiplication at every lag;
- this cost will not increase further if more columns / rows are shifted into the lag-zero matrix at every iteration



- MS-SMD will transfer yet more off-diagonal energy per iteration;
- because the total energy must remain constant under paraunitary operations, SBR2, SMD and MS-SMD can be proven to converge.

Multiple Shift SMD (SMD)

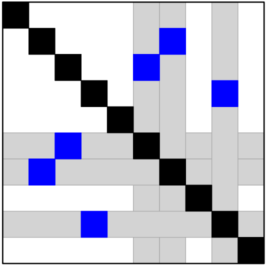
- SMD converges faster than SBR2 — more energy is transfered per iteration step;
- SMD is more expensive than SBR2 — full matrix multiplication at every lag;
- this cost will not increase further if more columns / rows are shifted into the lag-zero matrix at every iteration



- MS-SMD will transfer yet more off-diagonal energy per iteration;
- because the total energy must remain constant under paraunitary operations, SBR2, SMD and MS-SMD can be proven to converge.

Multiple Shift SMD (SMD)

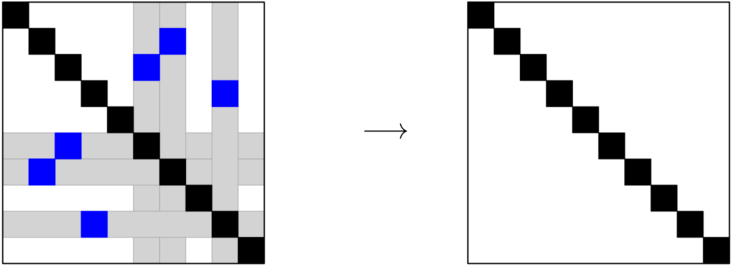
- SMD converges faster than SBR2 — more energy is transfered per iteration step;
- SMD is more expensive than SBR2 — full matrix multiplication at every lag;
- this cost will not increase further if more columns / rows are shifted into the lag-zero matrix at every iteration



- MS-SMD will transfer yet more off-diagonal energy per iteration;
- because the total energy must remain constant under paraunitary operations, SBR2, SMD and MS-SMD can be proven to converge.

Multiple Shift SMD (SMD)

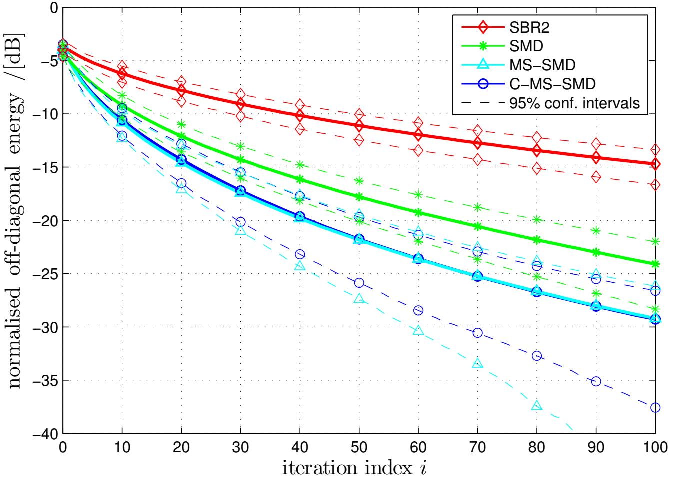
- SMD converges faster than SBR2 — more energy is transfered per iteration step;
- SMD is more expensive than SBR2 — full matrix multiplication at every lag;
- this cost will not increase further if more columns / rows are shifted into the lag-zero matrix at every iteration



- MS-SMD will transfer yet more off-diagonal energy per iteration;
- because the total energy must remain constant under paraunitary operations, SBR2, SMD and MS-SMD can be proven to converge.

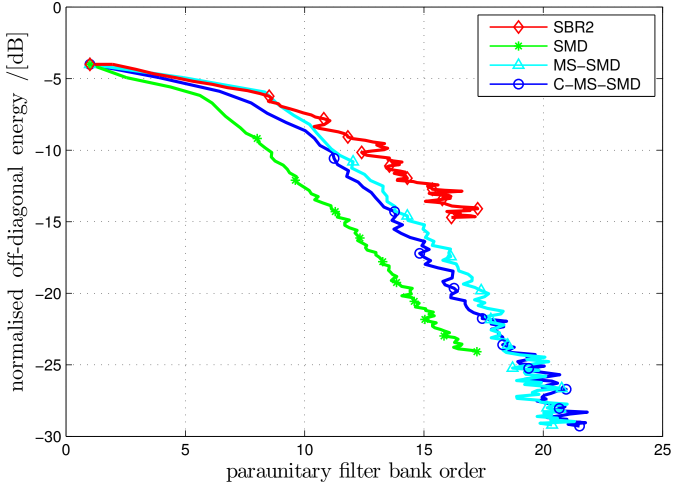
SBR2/SMD/MS-SMD Convergence

- Measuring the remaining normalised off-diagonal energy over an ensemble of space-time covariance matrices:



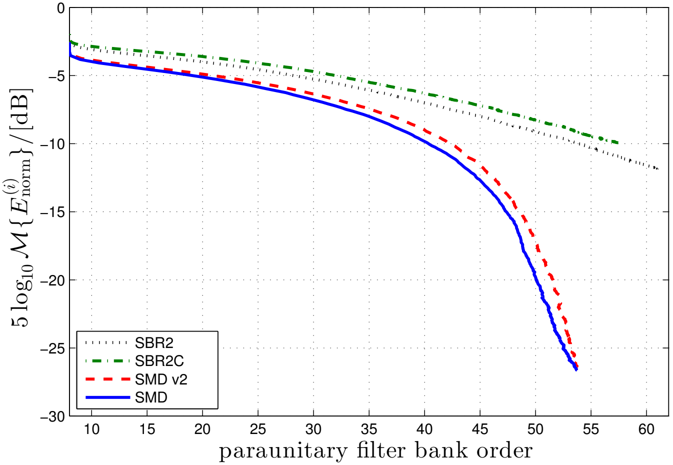
SBR2/SMD/MS-SMD Application Cost 1

- Ensemble average of remaining off-diagonal energy vs. order of paraunitary filter banks to decompose 4x4x16 matrices:



SBR2/SMD/MS-SMD Application Cost 2

- Ensemble average of remaining off-diagonal energy vs. order of paraunitary filter banks to decompose 8x8x64 matrices:



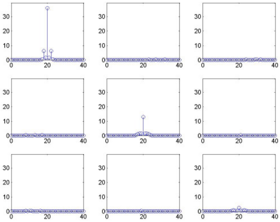
Polynomial MUSIC (PMUSIC)

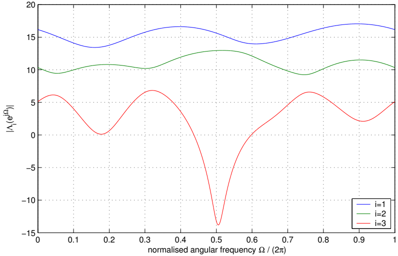
[Alrmah, Weiss, Lambotharan, *EUSIPCO* (2011)]

- Based on the polynomial EVD of the broadband covariance matrix

$$\mathbf{R}(z) \approx \underbrace{[\mathbf{Q}_s(z) \quad \mathbf{Q}_n(z)]}_{\mathbf{Q}(z)} \underbrace{\begin{bmatrix} \mathbf{\Lambda}_s(z) & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda}_n(z) \end{bmatrix}}_{\mathbf{\Lambda}(z)} \begin{bmatrix} \tilde{\mathbf{Q}}_s(z) \\ \tilde{\mathbf{Q}}_n(z) \end{bmatrix}$$

- paraunitary $\mathbf{Q}(z)$, s.t. $\mathbf{Q}(z)\tilde{\mathbf{Q}}(z) = \mathbf{I}$;
- diagonalised and spectrally majorised $\mathbf{\Lambda}(z)$:





PMUSIC cont'd

- Idea —- scan the polynomial noise-only subspace $Q_n(z)$ with broadband steering vectors

$$\Gamma(z,\vartheta) = \tilde{\mathbf{a}}_\vartheta(z)\tilde{\mathbf{Q}}_n(z)\mathbf{Q}_n(z)\mathbf{a}_\vartheta(z)$$

- looking for minima leads to a spatio-spectral PMUSIC

$$S_{\mathrm{PSS-MUSIC}}(\vartheta,\Omega) = \left(\Gamma(z,\vartheta)|_{z=e^{j\Omega}}\right)^{-1}$$

- and a spatial-only PMUSIC

$$S_{\mathrm{PS-MUSIC}}(\vartheta) = \left(2\pi \oint \Gamma(z,\vartheta)|_{z=e^{j\Omega}} d\Omega\right)^{-1} = \Gamma_\vartheta^{-1}[0]$$

with $\Gamma_\vartheta[\tau] \circ\!\!-\!\!\bullet\, \Gamma(z,\vartheta)$.

Simulation I — Toy Problem

- Linear uniform array with critical spatial and temporal sampling;
- broadband steering vector for end-fire position:

$$\mathbf{a}_{\pi/2}(z) = [1 \;\; z^{-1} \;\; \cdots \;\; z^{-M+1}]^{\mathrm{T}}$$

- covariance matrix

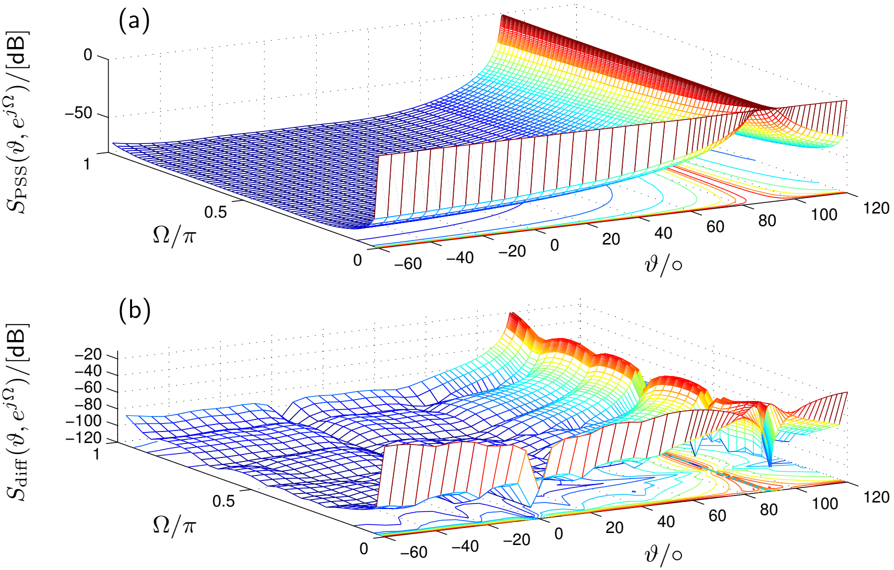
$$\mathbf{R}(z) = \mathbf{a}_{\pi/2}(z)\tilde{\mathbf{a}}_{\pi/2}(z) = \begin{bmatrix} 1 & z^{1} & \ldots & z^{M-1} \\ z^{-1} & 1 & & \vdots \\ \vdots & & \ddots & \vdots \\ z^{-M+1} & \ldots & \ldots & 1 \end{bmatrix} .$$

- PEVD (by inspection)

$$\mathbf{Q}(z) = \mathbf{T}_{\mathrm{DFT}}\mathrm{diag}\{1 \;\; z^{-1} \;\; \cdots \;\; z^{-M+1}\} \quad ; \quad \boldsymbol{\Lambda}(z) = \mathrm{diag}\{1 \; 0 \; \cdots \; 0$$

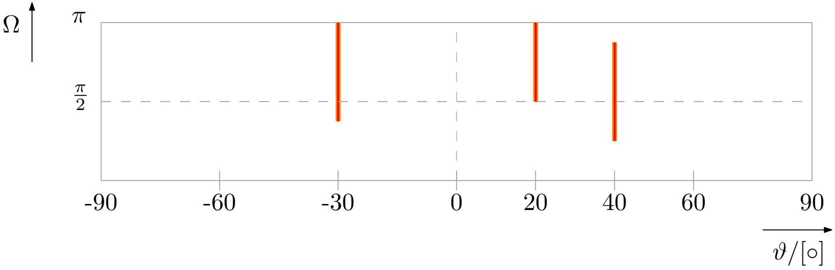
- simulations with $M = 4 \ldots$

Simulation I — PSS-MUSIC



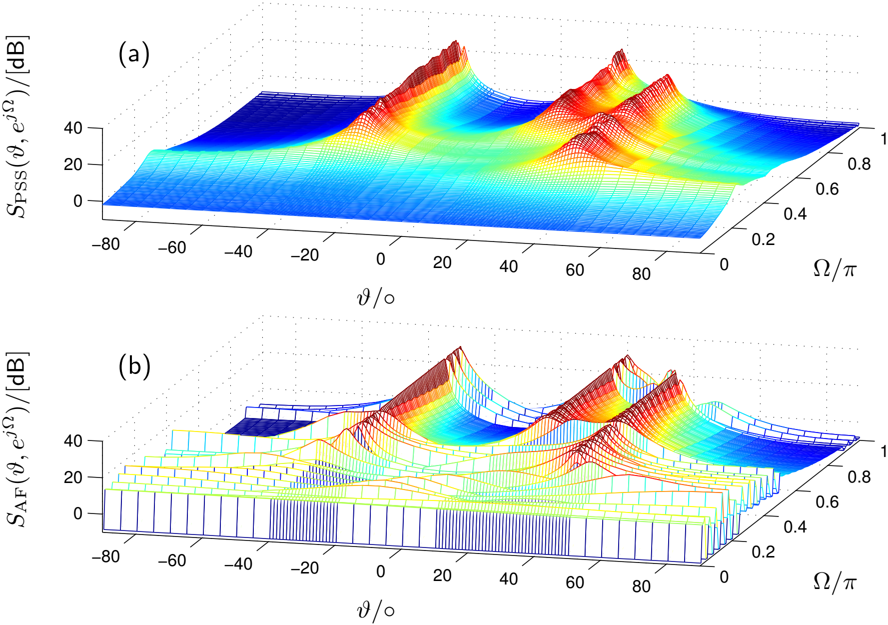
Simulation II

- $M = 8$ element sensor array illuminated by three sources;
- source 1: $\vartheta_1 = -30°$, active over range $\Omega \in [\frac{3\pi}{8};\ \pi]$;
- source 2: $\vartheta_2 = 20°$, active over range $\Omega \in [\frac{\pi}{2};\ \pi]$;
- source 3: $\vartheta_3 = 40°$, active over range $\Omega \in [\frac{2\pi}{8};\ \frac{7\pi}{8}]$; and



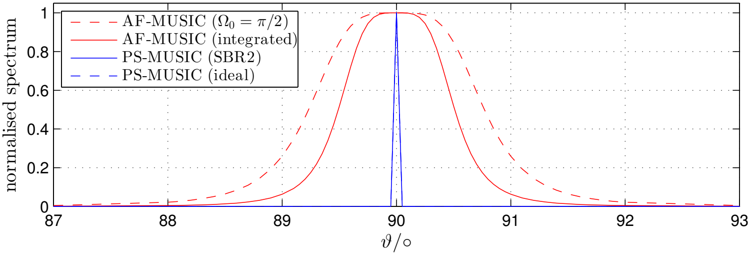
- filter banks as innovation filters, and broadband steering vectors to simulate AoA;
- space-time covariance matrix is estimated from 10^4 samples.

Simulation II — PSS-MUSIC

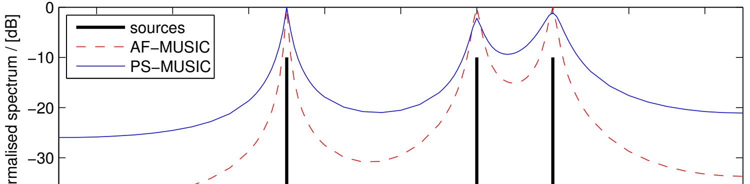


PS-MUSIC Comparison

- Simulation I (toy problem): peaks normalised to unity:



- Simulation II: inaccuracies on PEVD and broadband steering vector



Conclusions

- We have considered the importance of SVD and EVD for narrowband source separation;
- narrowband matrix decomposition real the matrix rank and offer subspace decompositions on which angle-of-arrival estimation alhorithms such as MUSIC can be based;
- broadband problems lead to a space-time covariance or CSD matrix;
- such polynomial matrices cannot be decomposed by standard EVD and SVD;
- a polynomial EVD has been defined;
- iterative algorithms such as SBR2 can be used to approximate the PEVD;
- this permits a number of applications, such as broadband angle of arrival estimation;
- broadband beamforming could then be used to separate broadband sources.

Additional Material

- Papers included on the USB drives:
  1. J.G. McWhirter, P.D. Baxter, T. Cooper, S. Redif, and J. Foster: "An EVD Algorithm for Para-Hermitian Polynomial Matrices," IEEE Transactions on Signal Processing, **55**(5): 2158-2169, May 2007.
  2. S, Redif, J.G. McWhirter, and S. Weiss: "Design of FIR Paraunitary Filter Banks for Subband Coding Using a Polynomial Eigenvalue Decomposition," IEEE Transactions on Signal Processing, **59**(11): 5253-5264, Nov. 2011.
  3. P. Baxter and J.G. McWhirter: "Blind signal separation of convolutive mixtures," Proc. 37th Asilomar Conference on Signals, Systems and Computers, **1**: 124-128, November 2003.
- If interested in trying the PEVD and its iterative algorithms yourself, please e-mail:
  - Jamie Corr (jamie.corr@strath.ac.uk), or
  - Stephan Weiss (stephan.weiss@strath.ac.uk)
- coming soon: Matlab-compatible toolbox with PEVD algorithms.